Architecture of Garnet Collaborative Portal

DoE Components Meeting
July 24 2001

Geoffrey Fox

IPCRES Laboratory for Community Grids

Computer Science, Informatics, Physics

Indiana University

Bloomington IN

gcf@indiana.edu

Garnet Heritage/Assumptions

- **Support Education, Training and if possible Computing as Grid(Web) Services**
- **Use best practice commercial and academic capabilities**
  - Access Grid, HearMe, Anabas, (JMS, WSDL, EJB, Castor, Oracle etc.)
  - Worry about Centra, WebEx, Placeware, Blackboard, WebCT, Saba, Groove, Docent etc.
  - Respect IMS/ADL Learning Object standards (http://www.adlnet.org) and GGF Computing Objects
- **Integrate Synchronous and Asynchronous ("learning management system") collaboration**
- **Support hand held and desktop clients (universal access)**

Garnet Technology

- **Uniform XML event (message) based architecture**
  - Linked with a publish-subscribe paradigm
- **XML Schema GXOS supports hierarchical data structures (compatible with DoD ADL SCORM for learning objects)**
  - XML for all metadata (Users, documents, computers) and object changes -- from text chats to display changes etc.)
- **Java Middleware using Enterprise Javabeans**
- **Production system uses JMS (Java Message Service) to implement publish-subscribe**
  - JMS does Synchronous and Asynchronous Messaging
- **MyXoS manages XML information nuggets**

Important Capabilities in Initial Garnet

- **Standard stuff: built in shared display, whiteboard, HearMe Audio control, quizzes, annotations, chat/IM (Jabber.org)**
  - Desktop video will be special case of shared display
- **Record and replay all features of session (SMIL)**
  - A/V, Presentation, Annotations, Text Chat
- **Several Specialized Collaborative Shared Export Viewers: JSP, (later HTML, Acrobat ..)**
- **Initial SVG (Scalable Vector Graphics) Shared Batik Viewer**
  - 2D Scientific Visualization/Whiteboard
  - Macromedia (Flash~SVG) and Adobe (already "all" to SVG)
- **Initial source of SVG: Convert PowerPoint VML/WMF to SVG**
  - Gives shared export model for PowerPoint with each client able to scale independently at high resolution

JMS (Java Message Service) Structure in Garnet

Basic primitive is a topic/property labeled queue = JXTA Pipe

Pipes are collections of either messages or other pipes
and just "nodes" in information hierarchy labeled by a URI

Event Receptor (Queue)

Subscribe

Subscribe

Publish

HHMS
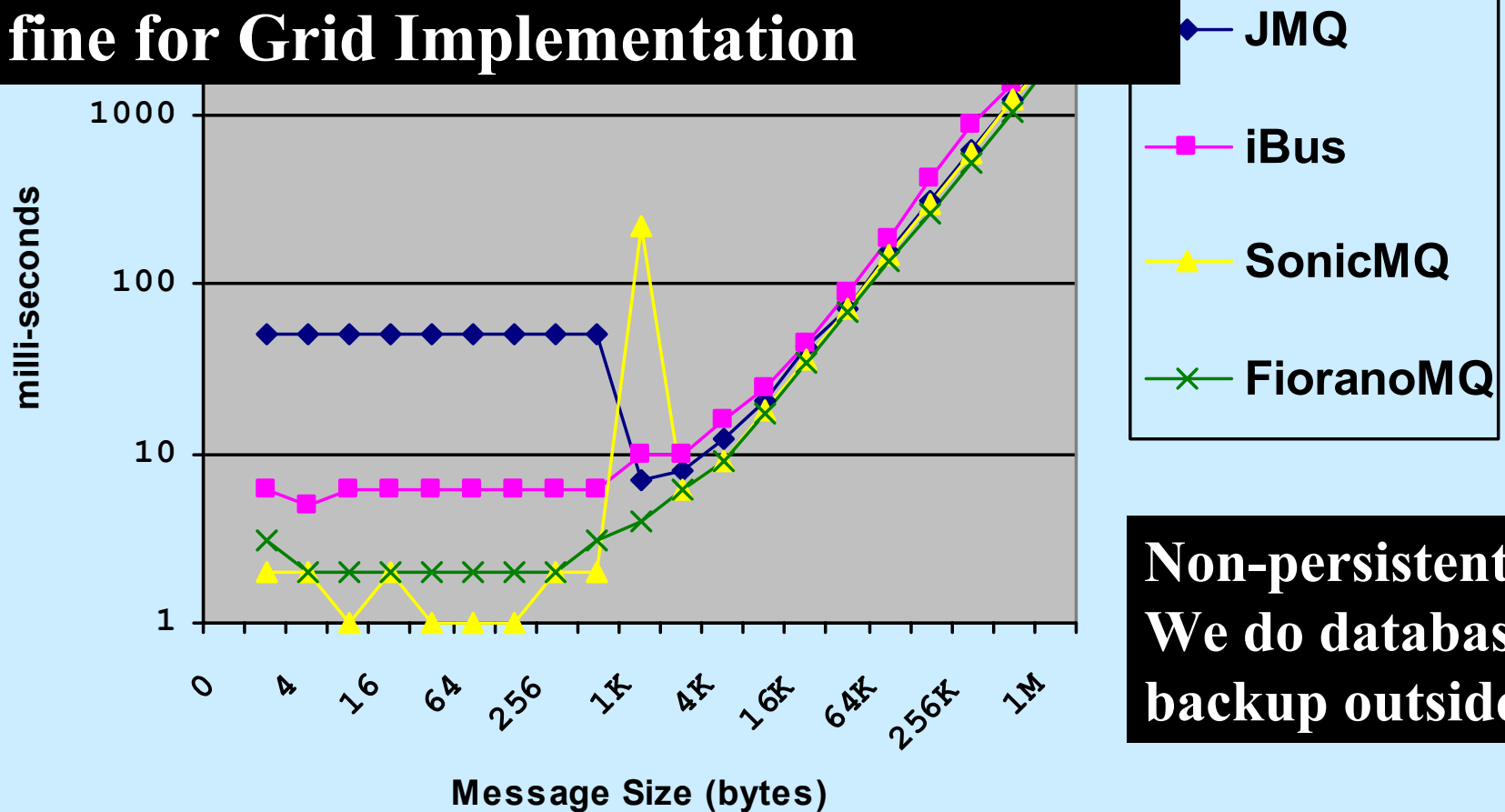
Convert Events to JMS

JavaScript

Java   C++   …...

HHMS (Hand Held Message Service) Optimized for Performance.

Performance of Commercial JMS I

One millisecond latency is fine for Synchronous Collaboration and fine for Grid Implementation
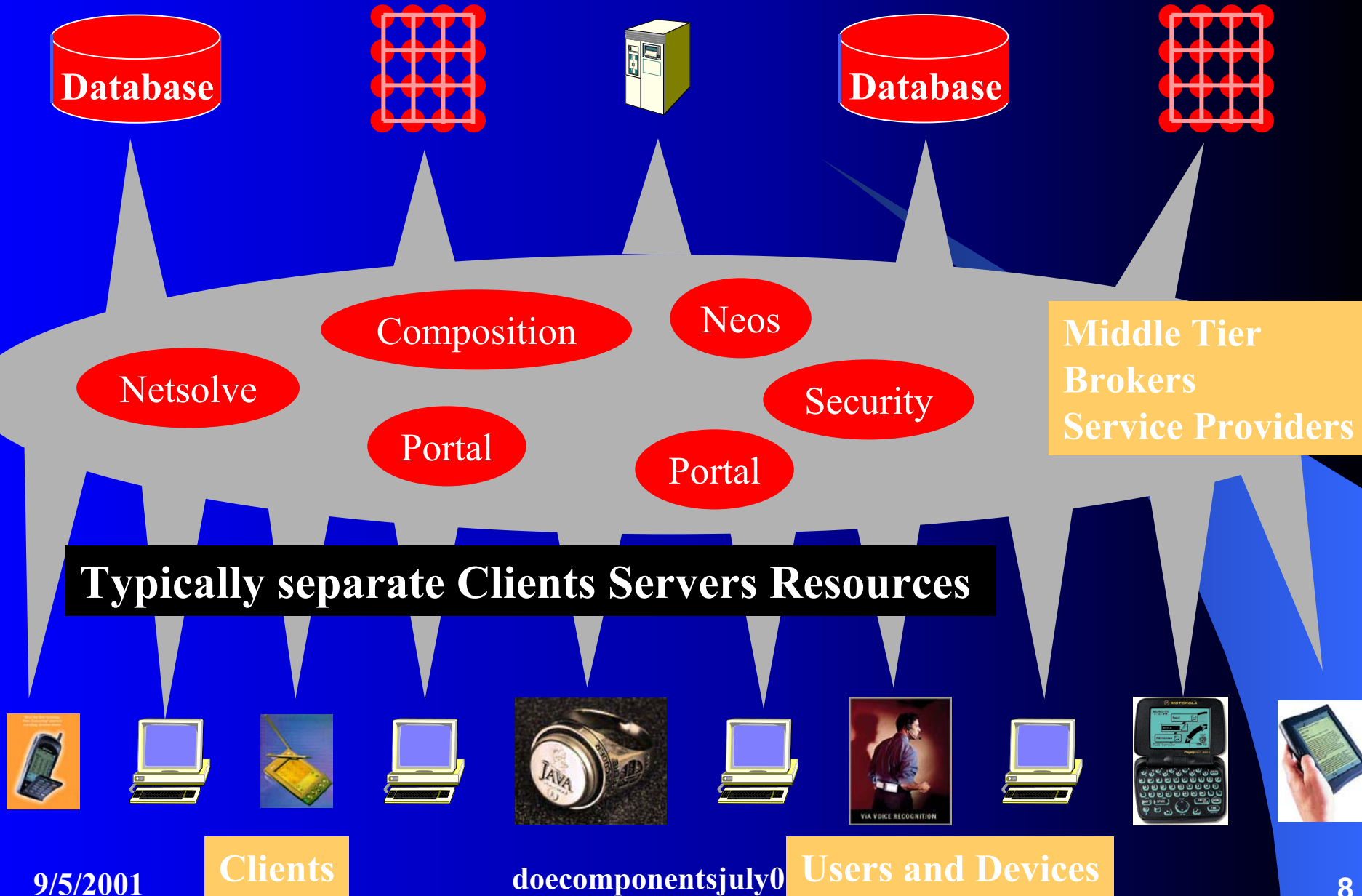

Non-persistent as We do database backup outside JMS

Peer to Peer P2P Networks

- **Publish/Subscribe is mechanism we use to establish who gets what information for Collaboration and P2P and may be ALL Grid and ALL Web Services?**

- **Gnutella and JXTA are different implementations**

Message (or event) services underlie P2P Grids

  (new ones) of P2P information propagation

  – GMS can be built on top of JXTA or JMS architecture

- **JXTA like MyXoS identifies the implicit distributed operating (control messages/metadata) system**

  – Both have message queues as primitives

  – Both have Shell

  – Both use XML based messages

  – JXTA Advertisements are similar to GXOS metadata for objects

Classic Grid Architecture

Resources

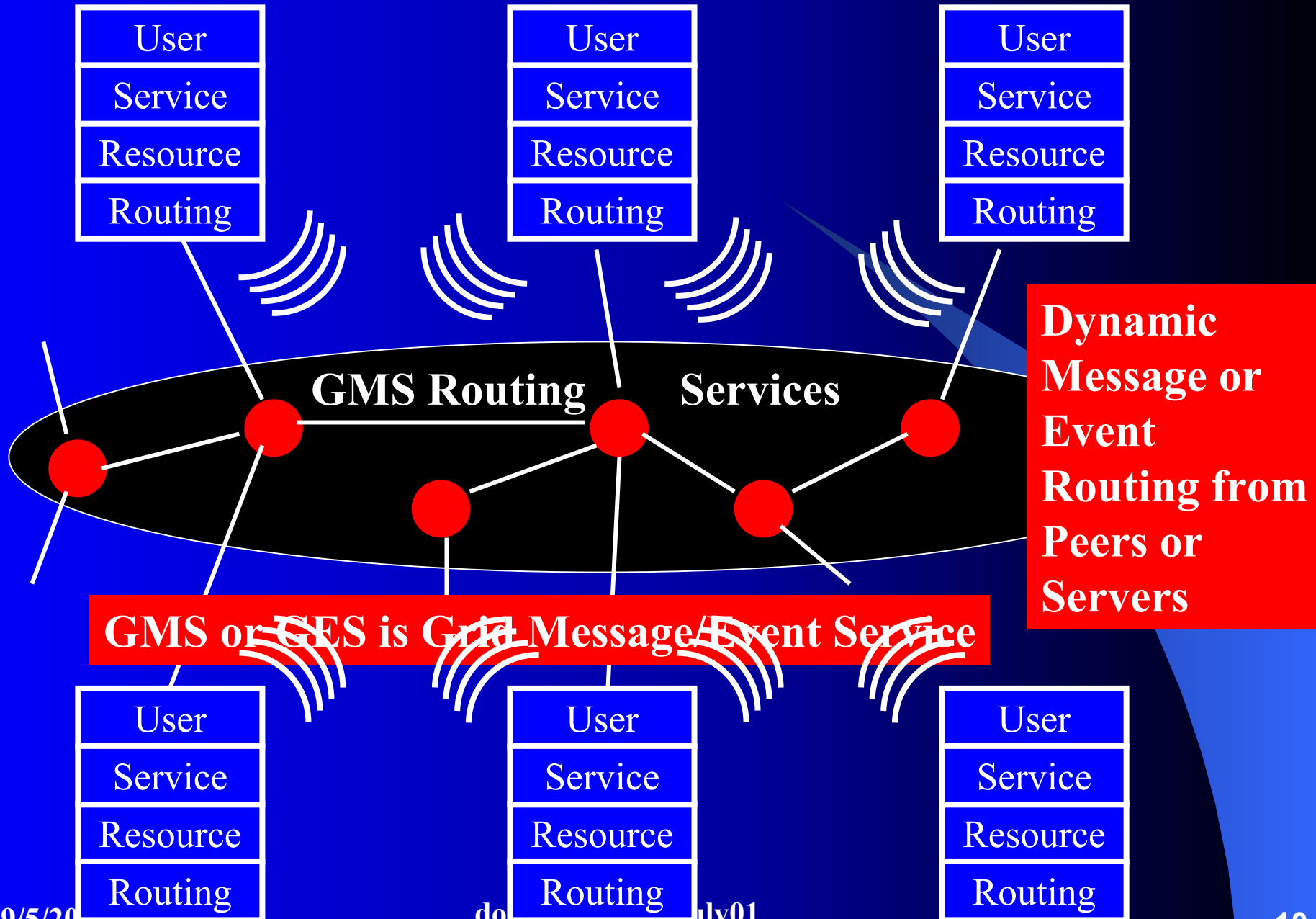**Database**

Database

Composition

Neos

Netsolve

Security

Portal

Portal

Middle Tier Brokers Service Providers

Typically separate Clients Servers Resources

Clients

doecomponentsjuly0

Users and Devices

Peer to Peer Network

Peers

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

Peers are Jacks of all Trades linked to "all" peers in community

Typically Integrated Clients Servers and Resources

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

Peer to Peer Grid

Dynamic Message or Event Routing from Peers or Servers

GMS Routing   Services

GMS or GES is Grid Message/Event Service

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

User
Service
Resource
Routing

Single Server P2P Illusion

Data base

JMS/GMS Server

Traditional Collaboration Architecture e.g. commercial WebEx and old Syracuse system Tango

doecomponentsjuly01

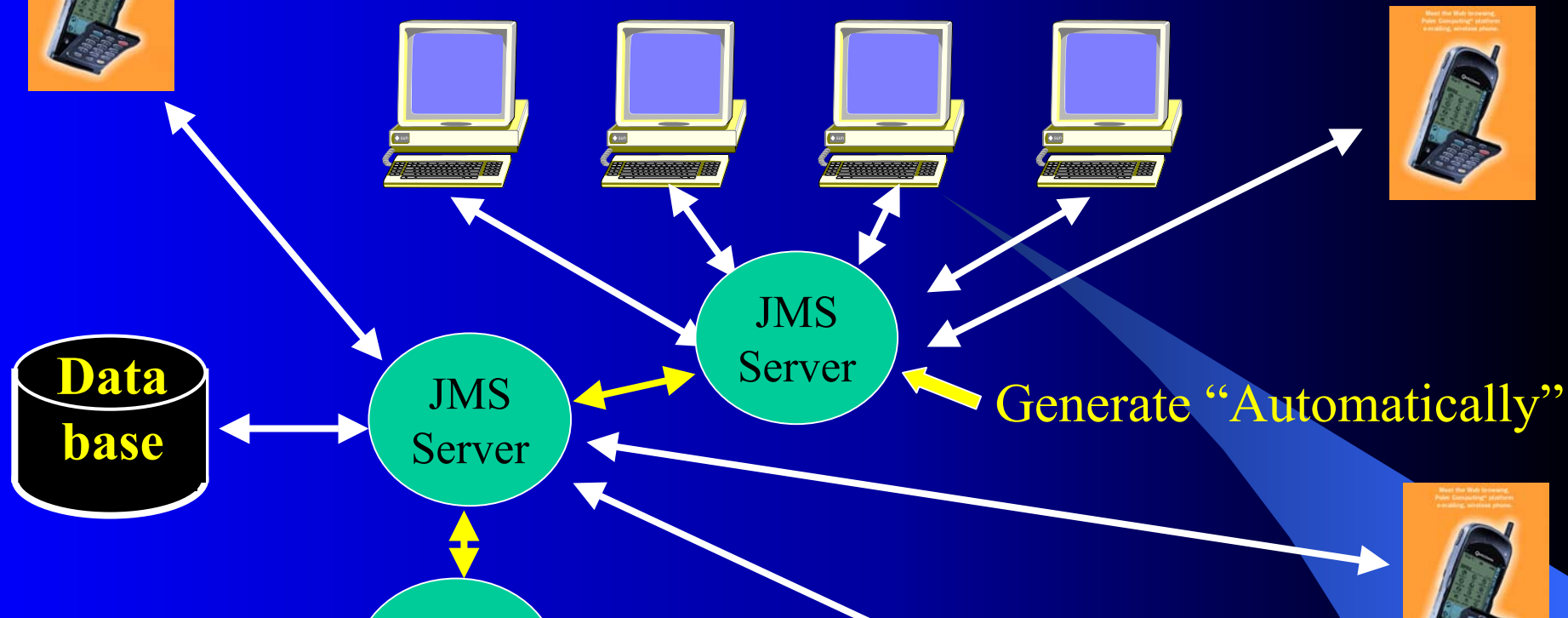# P2P Grid Event Service – a better JMS

- **Dynamic Collection of some billion computers each of which can either generate, route or consume events**

- **Publisher labels events by an (XML) object which is at simplest a URI but in general a collection of tag-values or instance of XML Schema**

- **Subscribers issue some sort of XML Query e.g. deliver all *gxos://garnet/Education/Graduate/ComputerScience/ Indiana/Spring2001/CPS616/Lecture3/***

- **Need Secure, High Performance, Efficient (don't propagate events further than they need), Fault Tolerant delivery service**

- **Shrideep Pallickara PhD June 1 2001**

- **Current version Java RMI based – could be SOAP**
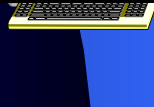
Proposed GMS Model for Messages

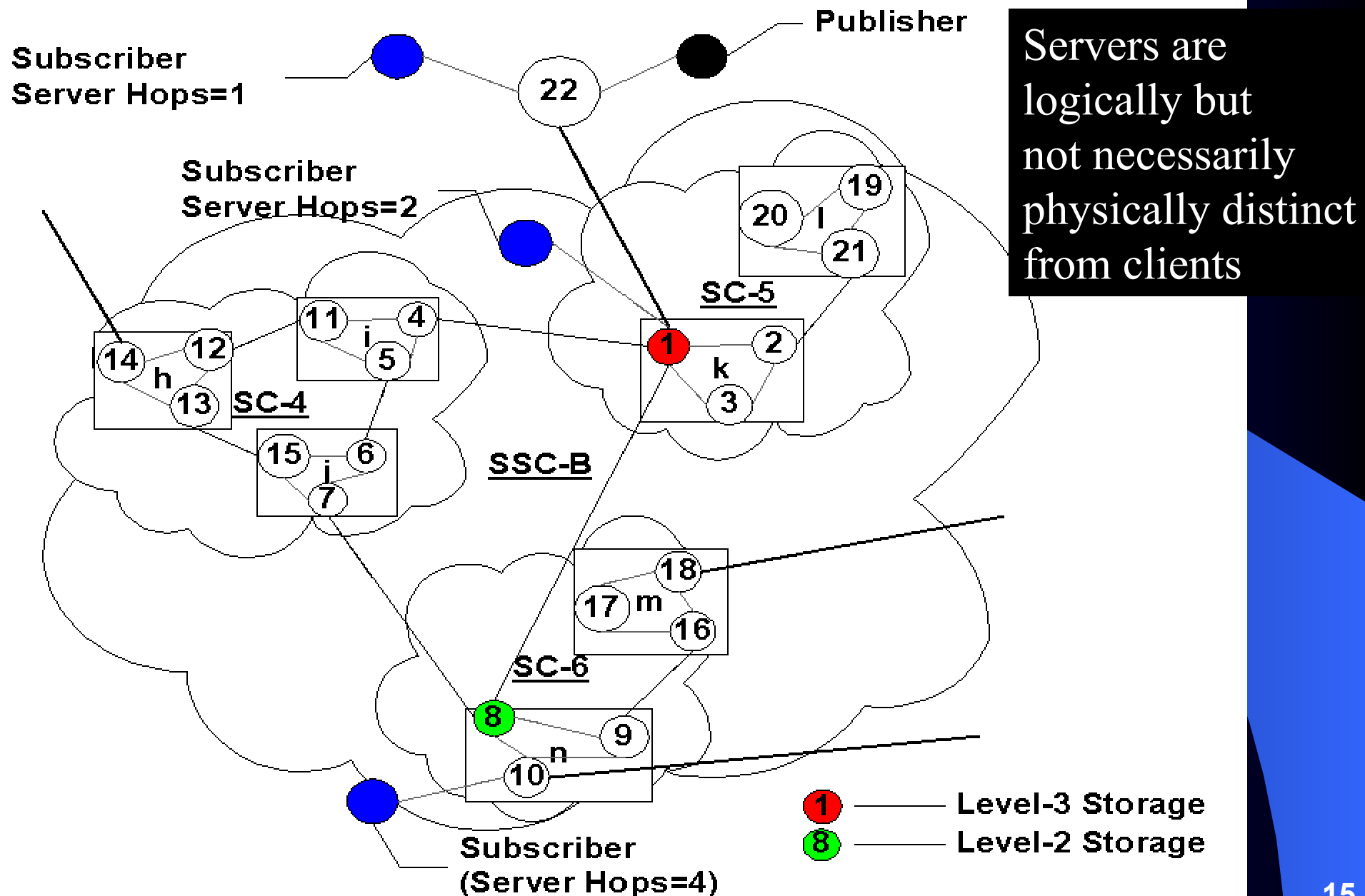- **All message publication labels and subscription profiles are defined in XML**

Database

Subscribes to all events to get persistence

Message Queue Labeled by (XML) Topic Object

Subscriber Profile Objects Specify Query to Event Label

Publishers

Subscribers
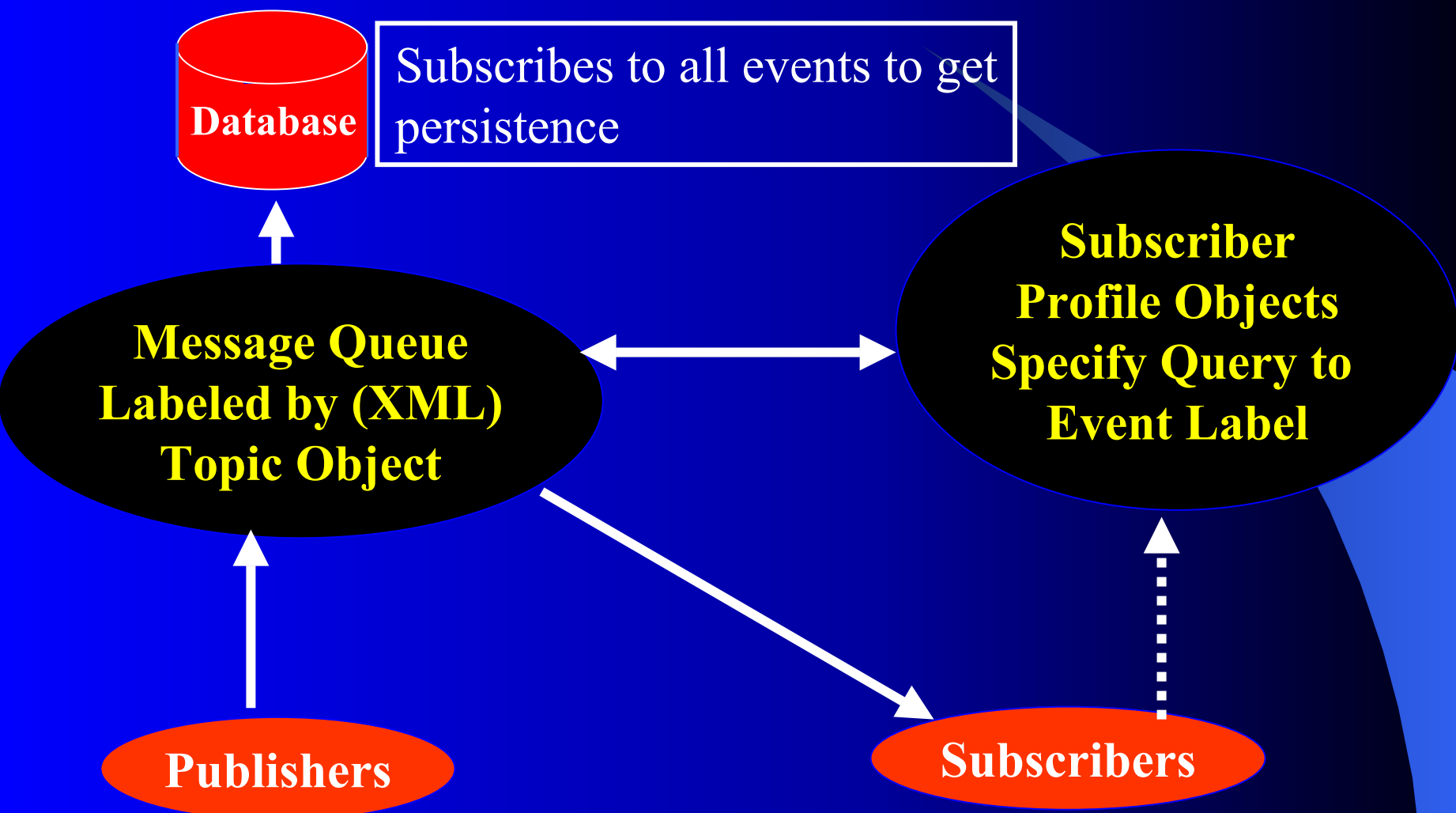
Multiple Server P2P Illusion

Data base

JMS Server

JMS Server

Generate "Automatically"

We are moving from client – server – resource model with
Clearly defined responsibilities to a
Heterogeneous Dynamic Grid of service providers and
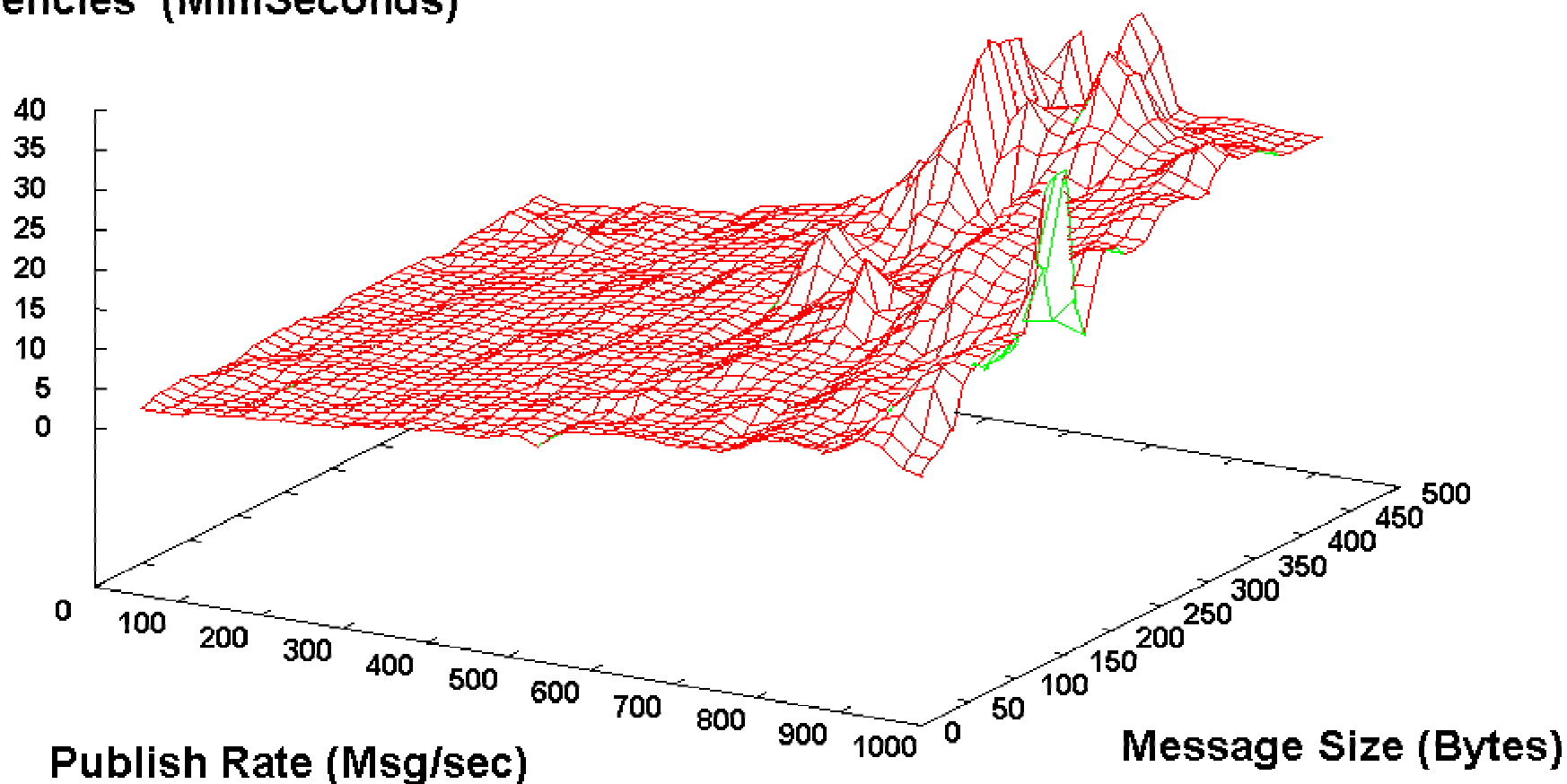Service consumers which are not necessarily distinct

Subscriber Server Hops=1

Publisher

Servers are logically but not necessarily physically distinct from clients

Subscriber Server Hops=2

SC-5

SC-4

SSC-B

SC-6

Subscriber (Server Hops=4)

Level-3 Storage
Level-2 Storage

Subscriber 1 server hop from publisher - Matching 10%

Event-based Garnet Architecture

- **All Objects are defined in XML** (metadata)– this XML view could be "virtual" but can be used to discover, edit (etc.) objects – labeled by a **URI**

- **GXOS** manages meta data defining all Objects -- it doesn't really want to manage Objects, just information required to **find, access, store, render and share** it

  – MUST have a good object management system to build collaboration service

  – Rendering includes Palm devices as well as PC's

  – Entities are people, cuuricula, grades, computers etc.

- **All actions including object changes are events** – all events are GXOS objects

  – Instant Messenger access, Framebuffer changes etc. are all GXOS events uniformly routed/archived etc.

- **There is a Shell MyXoS with basic Services (copy, create, collaborate etc.) – similar in concept to JXTA Shell**

Overall Structure of GXOS for a MegaMeeting

Capabilities

Global Root

Event Archive

Users

Devices

Documents

Admin

Multimedia
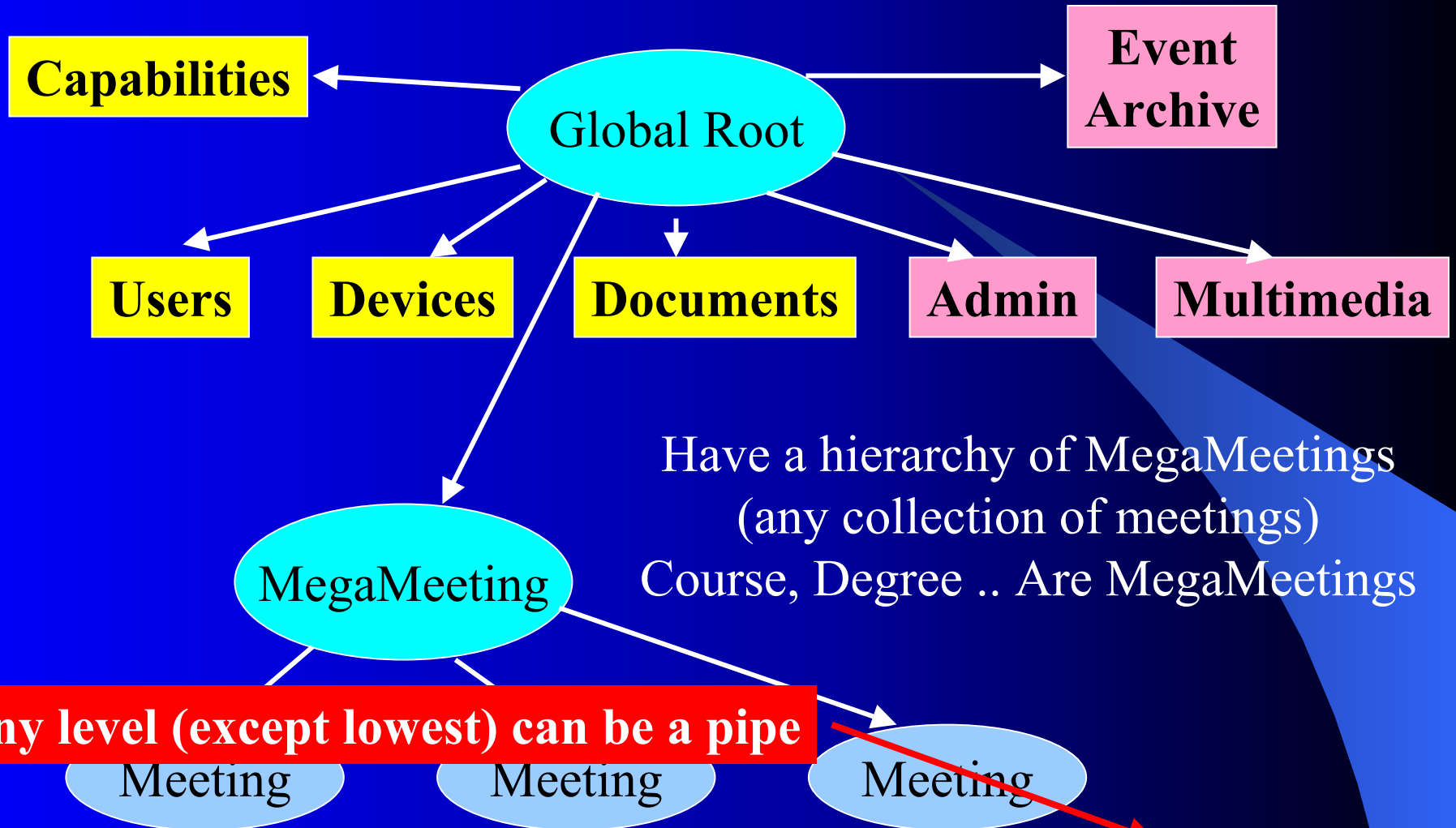
MegaMeeting

Have a hierarchy of MegaMeetings
(any collection of meetings)
Course, Degree .. Are MegaMeetings

**Any level (except lowest) can be a pipe**

Meeting

Meeting

Meeting

gxos://Education/University/Indiana/CS/PhD/Course/Lecture

Interface of XML and Java I

- **How will we teach computing?**
  - K-4: Internet Access
  - Middle School: (Simple) XML Schema interfaced to some scripting language
  - High School: Java as the programming model with Java classes (for external data) generated
- **Probably don't want to specify objects twice**
  - Start in Java; generate Schema
- **Or Start with Schema and generate Java**
- **Need a natural API of computer code to real or virtual XML**
  - Current mechanisms seem quaint (JDBC), inefficient (parsers), or non standard (Castor)

Interface of XML and Java II

- **Suppose we have a quadrillion (10^{15}) XML objects as say produced by a physics accelerator per year (Enterprise GXOS)**

- **Need to combine:**
  - **Search Interface to select nodes of XML Tree**
    - **Specify URI**
    - **JDBC or Google like Interfaces**
  - **Castor like Interface to map XML into Java but need to control depth of conversion from XML into Java**

Database

**(Virtual) XML View**

Choose And Convert

Middleware(EJB)
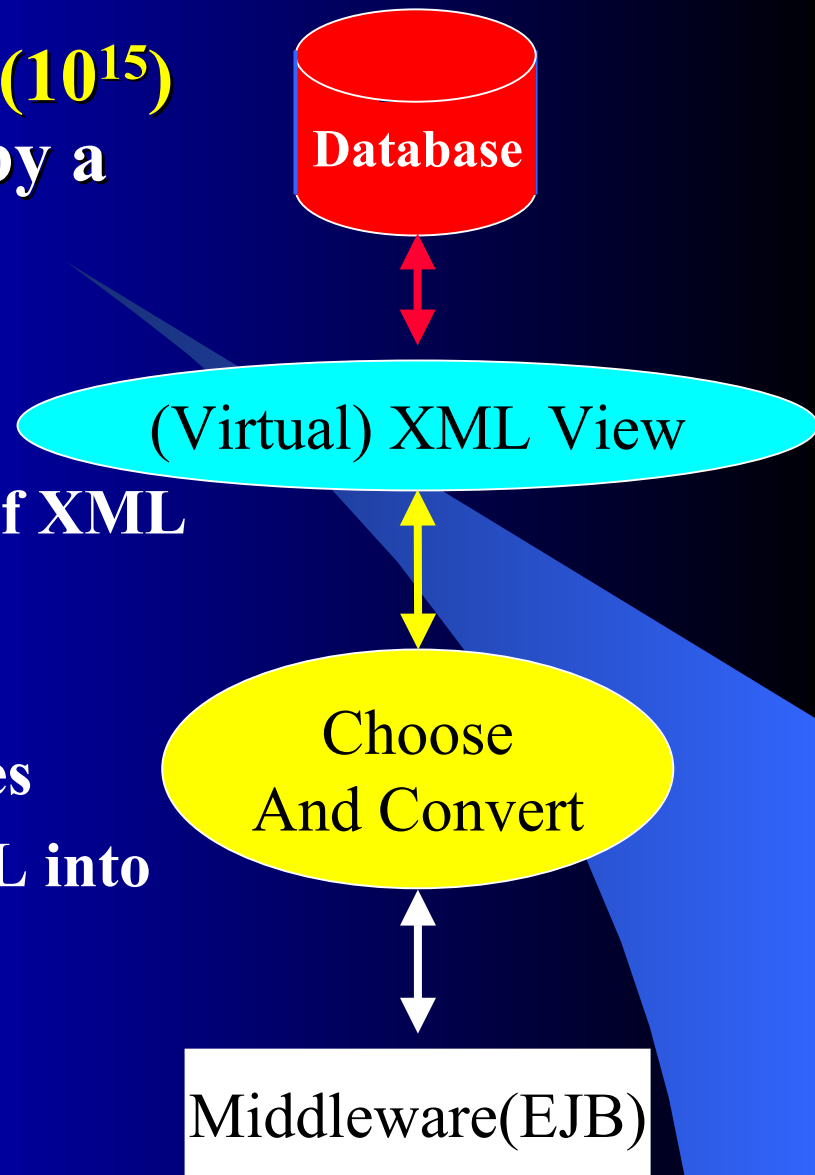
Current GXOS API Architecture

- **Initially implement "Personal GXOS" – Information Repository small enough that we can afford to read all possibly relevant information into memory and refine this**

  - E.g. Support course data for individual faculty

- **File.xml → XML Object → Java Object and vice versa**

  - Use Castor to automate XML Schema to Java Object

- **Primitives Supported Initially**

  - **Get a "leaf Object"**

  - **Get a Collection (Internal Node) – "handle" and self.xml (the GXOS properties associated with this node)**

  - **List Contents of a collection (recursively)**

  - **Get Contents of a collection (recursively)**

Integration of Hand Held Clients

- **Client Device** (machine) Profile stored in GXOS specifies O/S, default Screen Size modified by user (person) preferences
  - Dynamically updated with connection bandwidth, user updates
- **Application** Profile stored in GXOS and modified by event stream specifies data delivered by GMS
- **Adaptor (Personal Server)** looks like a conventional client to GMS and adapts data to specified client/user specifications
  - If PDA "small", then SVG viewer on "adaptor" and ship framebuffer to PDA
  - "Resize" on PDA handled by adaptor, scrolling by PDA etc.
  - Adaptor can process complex XML queries
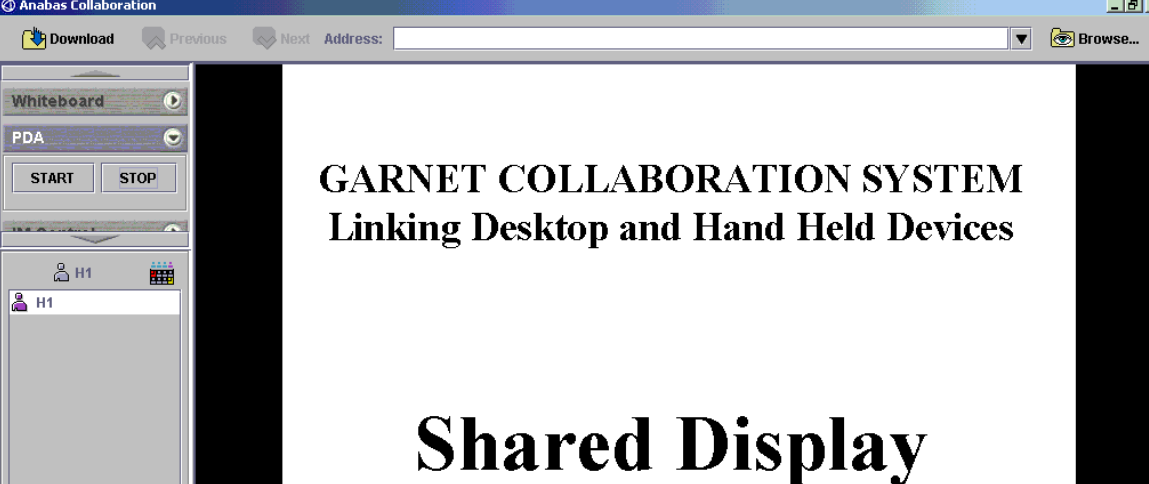
PowerPoint Shared Display PC to PDA